

Enhancing Chain-of-Thought Reasoning with Critical Representation Fine-tuning

Anonymous ACL submission

Abstract

Representation Fine-tuning (ReFT), a recently proposed Parameter-Efficient Fine-Tuning (PEFT) method, has attracted widespread attention for significantly improving parameter efficiency by editing representation space alone. In this work, we investigate applying ReFT to complex reasoning tasks. However, directly using the native ReFT method, which modifies fixed representations at the beginning and end of each layer, yields suboptimal performance, as these fixed-position representations have uncertain impact on the outputs. We observe that, in complex reasoning tasks, there often exist certain critical representations. These representations either integrate significant information from preceding layers or regulate subsequent layer representations. Through layer-by-layer propagation, they exert a substantial influence on the final output. Naturally, fine-tuning these critical representations has the potential to greatly enhance reasoning performance. Building upon these insights, we propose **Critical Representation Fine-Tuning** (CRFT), a novel method that identifies and optimizes these critical representations through information flow analysis. CRFT operates within a supervised learning framework, dynamically optimizing critical representations in a low-rank linear subspace while freezing the base model. The effectiveness and efficiency of our method are validated across eight benchmarks for arithmetic and commonsense reasoning, using LLaMA and Mistral model families. Notably, our method improves the accuracy of LLaMA-2-7B and ReFT by 18.2% and 3.8%, respectively, on GSM8K, while using only 0.016% of the model parameters, significantly less than other PEFT methods. Furthermore, our method also adapts effectively to few-shot settings, boosting one-shot accuracy by 16.4%. Our work highlights the untapped potential of representation-level optimization for CoT reasoning, offering a lightweight yet powerful alternative to traditional PEFT methods.

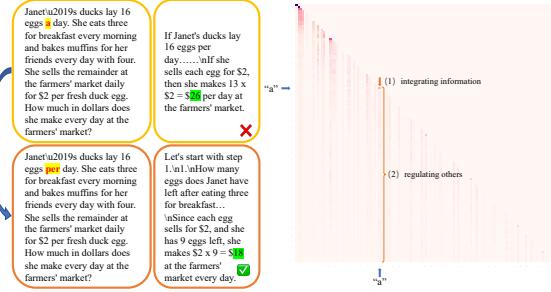


Figure 1: Examples of modifying a critical representation in the first layer (an input token). This example, conducted on LLaMA-2-13B, illustrates (1) two strategies of identifying critical representations and (2) the impact of modifying these representations on the output.

1 Introduction

Large language models (LLMs) have made significant advances in the treatment of complex reasoning tasks (Chu et al., 2023; Yao et al., 2024; Besta et al., 2024), which demand intricate logical reasoning and comprehensive explanations. These tasks differ from simpler in-context tasks that mainly involve straightforward information retrieval or classification. A pivotal element in these advancements is the Chain-of-Thought (CoT) (Wei et al., 2022), decomposing the reasoning process into several intermediary steps, particularly used in the domains of arithmetic (Lu et al., 2022; Imani et al., 2023; Lightman et al., 2023) and commonsense (Trinh and Le, 2018; Ling et al., 2017; Patel et al., 2021).

Representation Fine-Tuning (ReFT) (Wu et al., 2024b) has emerged as a promising approach, offering parameter efficiency by operating at the representation level. Representations are considered fundamental as they reveal the inner reasoning processes of large language models (LLMs). However, ReFT yields suboptimal performance in complex reasoning tasks, due to its reliance on altering fixed representations at the beginning and end of each layer, coupled with the unpredictable effects these

changes have on the output. Through empirical analysis, we observe that in complex reasoning tasks, certain critical representations exist within each layer, as illustrated in Figure 1. These representations either aggregate significant information from the previous layer or modulate other representations in the subsequent layer. Through layer-by-layer propagation, they exert a substantial influence on the final reasoning output. To further validate their importance, introducing random perturbations (0.01 Gaussian noise) to a random representation in each layer of LLaMA-2-7B on GSM8K resulted in a 1.4% accuracy drop, underscoring the sensitivity of model performance to these representations. Naturally, fine-tuning these critical representations holds significant potential to enhance reasoning performance. Building upon these insights, we propose a novel PEFT method termed **Critical Representation Fine-Tuning (CRFT)**.

We employ information flow analytics (Wang et al., 2023), utilizing attention and saliency scores (Simonyan, 2013) as explicit indicators to identify critical representations. Specifically, for representations that aggregate significant information from the preceding layer, we prioritize those with predominant self-information flow, as they effectively consolidate gathered information. For representations that modulate subsequent layers, we focus on those with substantial outgoing information flow, reflecting their regulatory influence, reflecting their significant regulatory influence. However, optimizing critical representations poses a significant challenge due to their context-dependent nature. While some representations positively contribute to outputs and require no optimization, others adversely affect performance, with necessary adjustments varying across contexts. To address this, we introduce adaptive learning within a supervised framework. Building on recent advances in parameter-efficient fine-tuning (PEFT) at the representation level (Wu et al., 2024b,a), we freeze the base model and optimize critical representations by learning updated directions in a low-rank linear subspace.

We conducted comprehensive experiments on eight reasoning datasets in two scenarios: arithmetic and commonsense (Talmor et al., 2018), using four base models covering the LLaMA and Mistral families. The experimental results demonstrate the effectiveness of our intervention. Specifically, our method achieves improvements of 18.2% over LLaMA-2-7B on the GSM8K dataset with only the

0.016% parameters of the model. Furthermore, our method can be easily extended to few-shot learning, achieving increases of 16.4% and 9.8% in one-shot and two-shot learning, respectively. Our work highlights the untapped potential of representation-level optimization for CoT reasoning, offering a lightweight yet powerful alternative to traditional prompt-centric and weight-centric methods.

2 Method

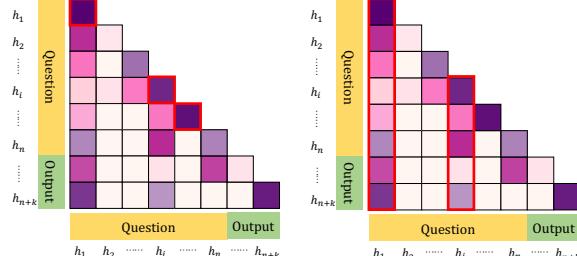
Our method, CRFT, consists of identifying and optimizing critical representations. We begin by introducing the problem formulation in Section 2.1. Next, we propose two strategies for identifying critical representations by analyzing the information flow, as presented in Section 2.2. Finally, we describe the way of optimizing critical representations in Section 2.3.

2.1 Problem Formulation

Given a sequence of n input tokens $\mathbf{x} = (x_1, \dots, x_n)$, the language model commences by embedding these tokens into a list of representations $\mathbf{h}^{(0)} = (\mathbf{h}_1^{(0)}, \dots, \mathbf{h}_n^{(0)})$. Since the vast majority of state-of-the-art language models are currently constructed based on the transformer (Vaswani, 2017) architecture, we focus solely on this architecture, which consists of L layers of transformer blocks. Subsequently, the L layers successively compute the l -th list of hidden representations $\mathbf{h}^{(l)}$ as a function of the previous list of hidden representations $\mathbf{h}^{(l-1)}$. Each hidden representation is a vector $\mathbf{h} \in \mathbb{R}^d$. Finally, the model leverages the last layer of hidden representations $\mathbf{h}^{(L)}$ to produce its predictions. Specifically, as a reasoning task, the model incrementally produces k tokens following the probability expression $p(x_{n+k}|x_1, \dots, x_n, x_{n+1}, \dots, x_{n+k-1})$. Our method aims to improve accuracy by identifying and optimizing critical representations $M(\mathbf{h})$.

2.2 Identify Critical Representations

Previous representation editing works also involve modification of representations but depend on empirical observations or general knowledge to locate representations for editing, which limits their adaptability and performance. For example, ReFT (Wu et al., 2024b) requires training and testing on other datasets to determine the optimal number of continuous representations to edit, specified as the first x and last y representations. This selection



(a) Self-referential filtering. (b) Multi-referential filtering.

Figure 2: The illustration of self-referential filtering and multi-referential filtering. We use red boxes to highlight the diagonal cells in Figure 2a and the column averages in Figure 2b that exceed the threshold α . The corresponding representations are marked with red lines and are referred to as critical representations.

process is not only cumbersome, but also lacks interpretability. Our work identifies critical representations $M(\mathbf{h})$, which significantly influence reasoning abilities and output correctness.

$$\begin{aligned} \mathbf{M}(\mathbf{h}) = \{ \mathbf{h}_i &| \text{Is correct}(\text{model}(\mathbf{h}_i + \epsilon)) \\ &\neq \text{Is correct}(\text{model}(\mathbf{h}_i)) \}, \end{aligned} \quad (1)$$

where ϵ is a small perturbation in a vector space. For simplicity, we use the abbreviation $\mathbf{M}^{(l)}$ to represent $M(\mathbf{h}^{(l)})$ in the following text. When all critical representations contribute positively to the output, accuracy is largely ensured.

As in the examples in Figure 1, whether a representation is a critical representation cannot be determined by itself but rather by its relationship with other representations. So, we utilize the information flow (Wang et al., 2023), leveraging attention and saliency scores as indicators. As shown in Figure 2, we use a grid to visualize the information interaction between representations, where cell (i, j) indicates the information interaction between representation j and representation i . The value of the cell (i, j) is indicated by attention scores or saliency scores, with darker colors signifying richer information interactions. The critical representations can be categorized into two functional roles: (1) integrating significant information from the preceding layer and (2) regulating the subsequent layer representations. Specifically, for the former, we focus on representations that consistently receive information flow from itself, indicating effective information accumulation. For the latter, we target representations that disseminate information to multiple others, indicating its rich information interaction. Consequently, we design two strategies

to filter critical representations: **self-referential filtering** and **multi-referential filtering**, respectively.

2.2.1 Self-Referential Filtering

If information from representation i mainly flows back to itself in the subsequent layer, it means that representation i contains important information or has effectively accumulated significant information. Consequently, we use $\text{Info}(i, i)$ as a critical metric to assess this retention. If $\text{Info}(i, i)$ is large, then $\text{Info}(i, j), j \neq i$ will be small since the values in a row are normalized through the softmax function. This situation suggests that the information flow from the representation i is predominantly directed toward itself, confirming that the representation i is indeed crucial.

$$M_{\text{diag}}^{(l)} = \{ \mathbf{h}_i^{(l)} \mid \text{Info}^{(l-1)}(i, i) > \alpha \}, i \in \{1, \dots, n\}. \quad (2)$$

To quantify information interactions, we employ attention scores and saliency scores as indicators, thus proposing two distinct ways: Self-Referential Attention Filtering (SAF) and Self-Referential Saliency Filtering (SSF), separately.

Self-Referential Attention Filtering (SAF). We utilize normal attention scores $A_i^{(l)}$, described in Eq. 3, as an explicit indicator to filter critical representations, since they quantify the relevance and degree of emphasis assigned to various representations within a sequence. This mechanism enables the model to dynamically concentrate on interactions and enhance its understanding capabilities.

$$\text{Info}_{\text{SAF}}^{(l)}(i, i) = A_i^{(l)} = \text{softmax}(\mathbf{h}_i^{(l)} (\mathbf{h}^{(l)})^T / \sqrt{d}), \quad (3)$$

Self-Referential Saliency Filtering (SSF). We also leverage saliency scores to filter critical representations. As saliency score is a widely accepted interpretation tool (Simonyan, 2013), comprehensively considers attention scores and gradient values, highlighting interactions from critical representations to the model output, as shown in Eq. 4,

$$\text{Info}_{\text{SSF}}^{(l)}(i, i) = A_i^{(l)} \odot \frac{\partial \mathcal{L}(x)}{\partial A_i^{(l)}}, \quad (4)$$

where \odot denotes the element-wise multiplication, and $\mathcal{L}(\cdot)$ represents the cross entropy loss function of the predicted probability distribution and the predicted class indices.

2.2.2 Multi-Referential Filtering

If information from representation j significantly affects multiple other representations, including

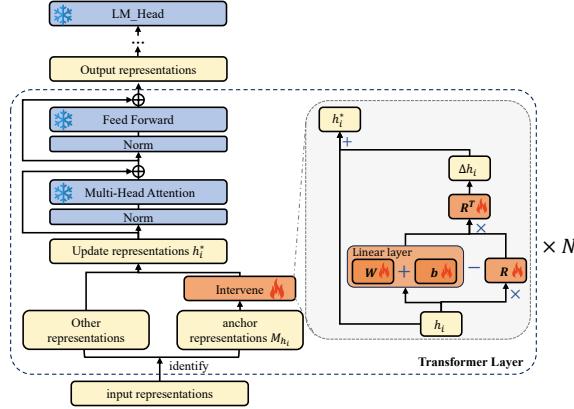


Figure 3: **The pipeline of optimizing critical representations.** Orange highlights the parameters to be learned, while blue indicates the parameters that remain frozen.

producing representations, then representation j is crucial. Specifically, we calculate the average of cells in the column j as a critical metric to represent the influence of j on other representations. If the average of $\text{Info}(\cdot, j)$ is large, then representation j has a substantial influence on others and plays a crucial role. As shown in Eq. 5, we use the threshold β to filter the critical representations,

$$M_{\text{col}}^{(l)} = \left\{ h_j^{(l)} \mid \frac{\sum_{i=j}^{n+k} \text{Info}^{(l)}(i, j)}{n + k - j + 1} > \beta \right\}, \quad (5)$$

where k is the number of output representations.

We also use the attention score and the saliency score to quantify the influence of representation j on representation i , which is termed Multi-Referential Attention Filtering (MAF) and Multi-Referential Saliency Filtering (MSF), respectively.

2.3 Optimize Critical Representations

Upon identifying critical representations, it becomes imperative to optimize them to ensure that their influence on reasoning tasks is accurately aligned. However, the direction of this modification remains uncertain and may not be unique. Consequently, we model the adjustment as a learnable vector Δh , which is learned during the training process to rectify the critical representations adaptively. Following (Wu et al., 2024b; Huang et al., 2024), we restrict our optimized vectors to a low-rank linear subspace employing a projection matrix with orthonormal rows $R \in \mathbb{R}^{r \times d}$, where r indicates the dimensionality of the subspace we are intervening in. We learn a projected source through a linear layer $\text{Linear}(h) = Wh + b$. Consequently, we modify the representation within the r -dimensional

Table 1: **Quantitative comparison of PEFT methods on GSM8K with LLaMA-2-7B.** The best performance is highlighted in **bold**, while the second-best is underlined.

PEFT	Param (%)	Identify	Accuracy (\uparrow)
None	-	-	14.6
LoRA (r=64)	0.826%	-	38.5
LoRA (r=8)	0.103%	-	36.7
RoSA ($r = 48$)	0.819%	-	30.5
RoSA ($r = 32$)	0.816%	-	32.2
RoSA ($r = 16$)	0.812%	-	32.8
SpA	0.809%	-	29.6
ReFT ($r = 8$)	0.031%	$p7 + s7$	29.0
		SAF	30.4 \uparrow 29.6
		MAF	32.0 \uparrow <u>32.1</u>
CRFT (ours)	0.016%	Union(attn)	31.2 \uparrow 32.8
		SSF	31.4 \uparrow 30.4
		MSF	31.4 \uparrow 30.3
		Union(sal)	32.8 \uparrow 31.5

subspace spanned by the rows of R to adopt the values derived from our linear projection source, $\text{Linear}(h)$. The overall optimization mechanism is depicted in Eq. 6,

$$\Phi(h) = \begin{cases} h + R^T(Wh + b - Rh), & \text{if } h \in M(h) \\ h, & \text{otherwise.} \end{cases} \quad (6)$$

3 Experiments

To validate the effectiveness of our method, CRFT, we performed experiments in two scenarios covering eight datasets: GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021), BoolQ (Clark et al., 2019), SocialIQA (Sap et al., 2019), WinoGrande (Sakaguchi et al., 2021), and OpenBookQA (Mihaylov et al., 2018). In particular, for the Commonsense task, previous work used the Commonsense170K dataset, which only provides the answers and lacks a reasoning process. We synthesized a Commonsense60K dataset with reasoning steps based on six commonly used commonsense datasets: CommonsenseQA (Talmor et al., 2018), CoS-e (Rajani et al., 2019), Open-BookQA (Mihaylov et al., 2018), SocialIQA (Sap et al., 2019), StrategyQA (Geva et al., 2021), WorldTree (Jansen et al., 2018). All experiments were conducted on the Pyvne (Wu et al., 2024c) codebase using a single GPU, either a NVIDIA A100 (80GB) or an L20 (40GB). And our method requires 4 hours for training on GSM8K with LLaMA-2-7B. We set the scoring method to the

Table 2: **Quantitative comparison on arithmetic and commonsense reasoning datasets with three base models: LLaMA-2-7B, LLaMA-3-8B, and Mistral-7B.** We train on Math10k and report results on AQuA, MAWPS, and SVAMP for arithmetic reasoning datasets; and we train on our combined commonsense datasets Commonsense60k and report results on four datasets: BoolQ, SocialIQA, WinoGrande, and OpenBookQA.

Model	PEFT	Identify	Accuracy (\uparrow)							
			AQuA	MAWPS	SVAMP	BoolQ	SocialIQA	OpenBookQA		
LLaMA-2-7B	CRFT (ours)	ReFT	p7+s7	21.7	80.7	52.2	50.7	61.2	51.7	58.6
		SAF	25.6 ± 26.0	78.6 ± 84.5	53.4 ± 52.6	60.0 ± 53.7	62.5 ± 67.4	60.6 ± 55.3	57.0 ± 62.2	
		MAF	27.6 ± 24.8	81.1 ± 80.7	52.4 ± 53.4	60.5 ± 61.8	52.8 ± 64.9	68.4 ± 51.8	50.6 ± 66.4	
		SSF	26.0 ± 26.8	80.7 ± 79.8	52.5 ± <u>53.3</u>	62.0 ± 54.3	<u>67.1</u> ± 64.4	60.2 ± 60.1	58.4 ± 58.6	
		MSF	<u>27.2</u> ± 22.8	79.4 ± 80.7	52.3 ± 52.5	60.0 ± 59.7	65.8 ± 63.4	54.5 ± 54.2	59.0 ± 56.4	
LLaMA-3-8B	CRFT (ours)	ReFT	p7+s7	46.9	87.0	74.2	62.1	60.2	56.0	66.0
		SAF	47.2 ± 47.2	89.9 ± 88.2	75.5 ± 76.1	63.0 ± 66.4	68.2 ± 67.1	62.6 ± 56.3	71.0 ± 73.6	
		MAF	48.4 ± <u>50.4</u>	90.8 ± 90.8	77.1 ± 77.9	62.4 ± 66.2	66.5 ± 62.7	67.2 ± <u>62.9</u>	73.8 ± 72.6	
		SSF	50.0 ± 49.2	86.6 ± 86.6	<u>78.0</u> ± 78.1	64.0 ± 66.6	74.7 ± 74.2	60.3 ± 62.0	75.6 ± 77.0	
		MSF	48.0 ± 51.6	87.0 ± 87.4	75.2 ± 74.8	<u>67.0</u> ± 67.9	67.4 ± 69.7	62.3 ± 62.8	70.0 ± 68.6	
Mistral-7B	CRFT (ours)	ReFT	p7+s7	32.3	84.9	67.4	62.5	64.6	58.5	63.8
		SAF	36.2 ± 38.6	87.0 ± 85.7	65.9 ± 66.2	63.0 ± 66.5	66.7 ± 75.6	61.5 ± 62.9	<u>72.6</u> ± <u>72.6</u>	
		MAF	<u>39.0</u> ± 38.2	84.9 ± 85.3	66.3 ± 65.3	62.1 ± 60.8	66.9 ± 71.5	<u>61.2</u> ± <u>63.7</u>	64.2 ± 69.6	
		SSF	37.4 ± 33.5	85.3 ± 84.5	<u>70.3</u> ± 70.6	62.3 ± 64.8	64.9 ± 62.9	64.3 ± 61.4	61.6 ± 66.4	
		MSF	41.3 ± 37.8	87.4 ± 85.3	66.0 ± 66.9	62.5 ± <u>65.0</u>	69.3 ± 71.8	62.3 ± 59.5	72.8 ± 68.6	

“order”, with α and β set both to 0.05. The ablation studies of these hyperparameters are discussed in Section 3.3. We adopt SAF strategy in Section 3.2 and Section 3.3. The details of all datasets and other implementations are reported in Appendix A. Our evaluation focused exclusively on the accuracy of the final numerical or multiple-choice answers. Generation examples are reported in Appendix C.

3.1 Quantitative Results

Table 1 summarizes the comparison of our method, CRFT, with other PEFT methods on GSM8k with LLaMA-2-7B. For each strategy, we report two accuracy values: the first value involves selecting critical representations by further filtering those identified as critical representations from the previous layer, while the second value focuses on identifying critical representations by filtering only within the current layer. Given that the optimal strategy may differ by context, we recommend a combined approach of self-referential and multi-referential filtering. Since the scoring systems of these two strategies are not directly comparable, the union of the filtered sets is employed. To ensure a fair comparison, the same number of critical representations is maintained, which may lead to the omission of some highly important ones. Consequently, the combined method may exhibit slightly lower performance in certain situations. Alternatively, adjusting the threshold α and β provides a solu-

tion: lowering α (β) increases interventions for improved performance, while raising α (β) decreases interventions for enhanced efficiency.

Without bells and whistles, our method is comparable with other PEFT methods with fewer learnable parameters. For example, one of our strategies, union with attention scores, outperforms LLaMA-2-7B and ReFT by 18.2% and 3.8%, respectively. Furthermore, the percentage of trainable parameters, calculated by dividing the trainable parameters by the total parameters of the model, highlights their substantial efficiency. Our method requires only 1/6 of the learnable parameters used by LoRA and 1/2 of those used by ReFT with the same rank.

Furthermore, our method, CRFT, consistently exhibits better performance on different models in arithmetic and commonsense scenarios. We report the results on different model sizes and model families on GSM8K: LLaMA-2-7B, LLaMA-2-13B, LLaMA-3-8B, and Mistral-7B, as shown in Appendix B. In addition, we present additional experimental results in arithmetic and commonsense scenarios, as shown in Table 2. We use the official public code of ReFT to report performance, as it only reports the results on LLaMA-1. And following the experimental conclusion of ReFT, we adopt the best intervention parameters $p7 + s7$, indicating the intervention in the first and the last seven representations. The consistent improvements observed in different reasoning tasks and different models under-

369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
Table 3: Expand our method to few-shot learning.

Few-shot	zero-shot	one-shot	two-shot
None	14.6	16.2	20.5
CRFT	29.6	28.7 ± 32.6	29.0 ± 30.3
Improvement	+15.0	+12.5 ± +16.4	+8.5 ± +9.8

Table 4: Ablation study on threshold α (β).

Threshold	1.0	0.25	0.05	0.01
Accuracy (↑)	24.7	30.0	29.6	33.2

score the robustness and versatility of our approach.

3.2 Expand to Few-shot Learning

Our method can easily be extended to few-shot learning. Intuitively, demonstrations should not directly affect the output; they are usually used to gain a higher-level semantic understanding, which then affects the output. However, representations in the question, such as numbers, can indeed have a direct impact. Consequently, we present experiments in Table 3 to examine whether the demonstration and the question should be learned independently. The first value suggests that the demonstration and the question are interdependent, leading to a single update vector for the critical representations. Conversely, the second value implies that the demonstration and the question are independent, resulting in distinct update vectors. These results prove the necessity of differentiating update directions between demonstrations and the question. Due to memory limitation, we only experimented with one-shot and two-shot.

3.3 Hyperparameter Configurations

We conducted extensive ablation studies on GSM8K using Llama-2-7B with the SAF strategy to systematically investigate hyperparameters, including the threshold α and β , the number of intervention representations and selection criteria.

The threshold α and β determines the degree to which the critical representations are. Given that the threshold values for α and β lie on the same dimension, we apply a unified threshold for both self-referential filtering and multi-referential filtering. We investigated four values, as shown in Table 4, and found that a threshold of 0.01 yields the best results. As the threshold decreases, the number of selected representations increases, but altering these selected representations can be more

Table 5: Ablation study on the number of intervention representations.

Number	0	14	20	30
Accuracy (↑)	14.6	29.6	30.3	27.7

Table 6: Ablation study on selection criteria.

Criteria	order	score	random
Accuracy (↑)	29.6	28.7	23.1

difficult. However, a lower threshold also carries the risk of excluding significant representations.

In the implementation, the number of intervention representations for each layer is fixed. If the number of critical representations obtained through the SAF strategy exceeds the number of intervention representations, we sample them using specific selection criteria. Conversely, if the number of critical representations is fewer than needed, we use a placeholder value of -1 to pad the length. For a fair comparison with the ReFT method, we set the default number of intervention representations to 14. An ablation study on the number of representations, shown in Table 5, revealed that the results were optimal when set to 20. When the number of intervention representations becomes too large, it hinders the learning of the update direction, leading to suboptimal results compared to using fewer representations.

For selection criteria, we designed three approaches to sample critical representations: positional order, score ranking, and random selection. The results, shown in Table 6, indicate that positional order selection is superior, while random selection yields significantly lower results compared to the other two criteria.

The ablation study presented above suggests that careful selection of hyperparameters can further enhance the performance of our CRFT method.

3.4 Are critical representations instrumental?

We validate that the selected representations are critical representations with a significant impact on the output. Using SAF as an identification strategy, we selected the top 5 and last 5 representations

Table 7: The validation of effectiveness in each layer.

Layer	None	0	31	0-15	16-31	all
Acc. (↑)	14.6	24.9	22.7	30.5	24.6	29.6

Table 8: **The necessity of identifying critical representations.** We present the results with LLaMA-2-7B on GSM8K. The best way to identify critical representations is highlighted in **bold**, while the second-best is underlined.

Location	None	ReFT	Our CRFT	Uniform Random										
		$(p7 + s7)$	(MAF)	37	38	39	40	41	42	43	44	45	46	47
Accuracy (\uparrow)	14.6	<u>29.0</u>	32.1	26.6	26.6	28.1	27.3	25.5	24.5	27.8	27.5	28.1	26.2	26.4

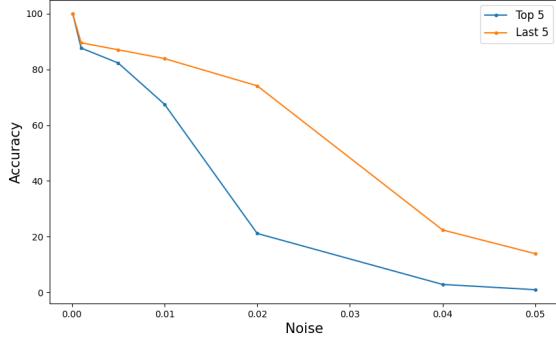


Figure 4: **The validation of critical representations identification.** Accuracy of originally correct examples under noise in the top 5 and last 5 representations.

based on their scores for each layer on GSM8K with LLaMA-2-7B. The effects of adding noise to these representations are presented in Figure 4, where the x-axis represents the magnitude of the noise, and the y-axis shows the proportion of originally correct examples remaining correct. We observe that the accuracy of the top 5 representations decreases rapidly with increasing noise. When the noise level is 0.02, the accuracy of the top 5 representations drops to 21.1%, whereas the last 5 representations maintain an accuracy of 74.1%. This result demonstrates the significant impact of critical representations on output performance.

In addition, we investigated the necessity of identifying critical representations. We tested random intervention locations using seed values ranging from 37 to 47. As shown in Table 8, the interventions of random representations during training can surpass the original LLaMA-2-7B, as the update direction is learnable. However, it remains inferior to the results achieved through our careful identification of critical representations, highlighting the necessity of this process.

Furthermore, we verified that intervention is necessary at each layer. As shown in Table 7, we intervene in the first layer (Layer 0), the final layer (Layer 31), the first half of the layers [0 – 15], the last half of the layers [16 – 31], and all layers. We found that each intervention improved accuracy and that interventions in the earlier layers have a

greater impact on the results. However, the best performance was achieved by intervening in the first half of the layers, as earlier feature representations are more closely aligned with the task objectives and can propagate throughout the model.

3.5 How do critical representations impact information flow?

We visualize attention maps to capture the variations in the information flow. The first and last heads in the final layer (Layer 32) of both LLaMA-2-7B and our proposed method, CRFT (SAF), are illustrated in Figure 5. A comprehensive comparison of all heads is provided in Appendix D, and the phenomenon is consistent. We have identified three observations, as follows:

- **Excessive information interaction in the representation h_0 is reduced.** In column 0, the absence of prominent color indicates a diminished influence of representation h_0 on other representations. The initial representation h_0 in LLaMA-2-7B lacks semantic information, but attracts a high level of attention. Previous works (Xiao et al., 2023; Yu et al., 2024) have referred to this phenomenon as the “attention sink”. By applying our method, the representation h_0 receives less undue attention, leading to a more balanced distribution of attention.
- **Increased information interaction between representations.** The increase in the number of vertical lines signifies a heightened interaction among the representations.
- **Broader information flow.** The presence of high attention scores along the diagonal has shifted from a few isolated peaks to multiple cells. This denotes a broader information flow from various representations.

Based on the visualized results above, our method alters the direction of information flow, guiding it towards a more optimal path, and enriching the overall information interaction.

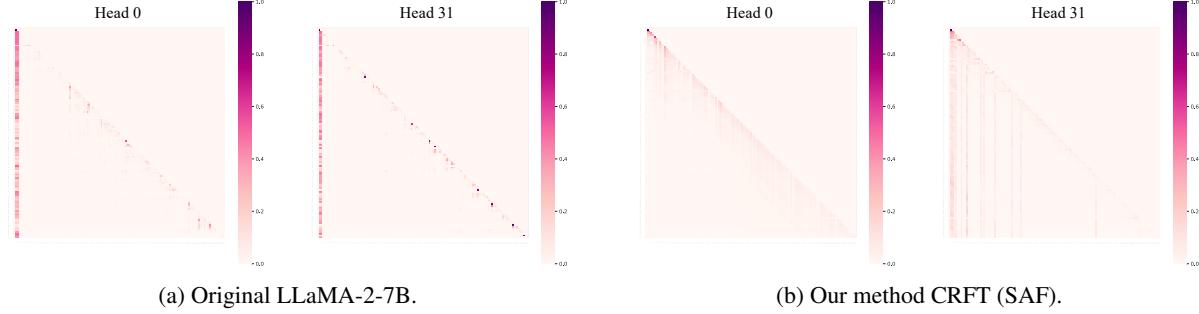


Figure 5: Visualization of attention scores for the first and last heads in the last layer.

4 Related Work

Intervention in LLMs. Intervention strategies encompass various techniques designed to influence the behavior of large-scale models during the inference phase. Common strategies include activation editing (Li et al., 2024), weight editing (Dai et al., 2022), and the use of guidance vectors (Zou et al., 2023), as well as altering the output distribution through comparative analysis (Li et al., 2022; Chuang et al., 2023). As representations encode rich information, some methods (Geiger et al., 2021; Wu et al., 2024b) change the output by editing representations. Although representation interventions can serve as powerful tools for model control, previous methods intervene in representations based on empirical observations (Wu et al., 2024a,b) or general knowledge (Zhang et al., 2023). The above approaches are not general and time-consuming, which limits their adaptability and performance. In contrast, our method precisely identifies the representations to intervene.

Information Flow Analysis. Recent studies have utilized attention mechanisms to analyze their impact on model performance. For example, Stream-LLM (Xiao et al., 2023) discovered that the initial token of an input text often receives an excessive amount of attention, despite frequently lacking semantic significance. It suggests that we should preserve these tokens when processing long input sequences to prevent forgetting. Furthermore, ACT (Yu et al., 2024) found that attention sinks can occur not only at the initial token but also throughout the entire sequence. Moreover, it was discovered that these attention sinks are not always beneficial to model performance. ACT optimizes attention distributions during inference, but not all heads can benefit from the calibration. Similarly, PASTA (Zhang et al., 2023) demonstrates that increasing the attention score of defined tokens in

specific heads can improve the ability of LLM to follow instructions. However, tokens need to be manually defined. Our method addresses these challenges by adaptively learning the updated direction of critical representations during training, leading to better overall performance.

5 Conclusion

We propose a novel Chain-of-Thought (CoT) reasoning method, termed Critical Representation Fine-Tuning (CRFT), which focuses exclusively on critical representations to influence model outputs. CRFT first identifies critical representations by analyzing the information flow through attention and saliency scores, and subsequently optimizes them via supervised fine-tuning within a low-rank subspace. Comprehensive experiments conducted across various models and datasets validate the effectiveness and efficiency, providing a new perspective on CoT reasoning tasks, particularly in long CoTs. Furthermore, CRFT exhibits sufficient flexibility to be readily adapted to few-shot learning scenarios, underscoring its potential to enhance reasoning capabilities within models.

Limitation

For identification, we currently focus on searching for representations that significantly impact the model output. However, it is important to note that representations with minor impacts may still have an influence, even if their effects are often negligible. A more effective strategy could involve prioritizing the correction of representations with negative impacts, although identifying such representations remains a challenge. Furthermore, while our optimizations are currently restricted to linear spaces, there is potential to explore alternative optimization methods that could enhance our framework.

References

- 587
588 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gersten-
589 berger, Michal Podstawski, Lukas Gianinazzi, Joanna
590 Gajda, Tomasz Lehmann, Hubert Niewiadomski, Pi-
591 ot Nyczek, et al. 2024. Graph of thoughts: Solving
592 elaborate problems with large language models. In
593 *Proceedings of the AAAI Conference on Artificial
594 Intelligence*, volume 38, pages 17682–17690.
- 595 Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang
596 Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu,
597 Bing Qin, and Ting Liu. 2023. A survey of chain of
598 thought reasoning: Advances, frontiers and future.
599 *arXiv preprint arXiv:2309.15402*.
- 600 Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon
601 Kim, James Glass, and Pengcheng He. 2023. Dola:
602 Decoding by contrasting layers improves factu-
603 rality in large language models. *arXiv preprint
604 arXiv:2309.03883*.
- 605 Christopher Clark, Kenton Lee, Ming-Wei Chang,
606 Tom Kwiatkowski, Michael Collins, and Kristina
607 Toutanova. 2019. Boolq: Exploring the surprising
608 difficulty of natural yes/no questions. *arXiv preprint
609 arXiv:1905.10044*.
- 610 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,
611 Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
612 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro
613 Nakano, Christopher Hesse, and John Schulman.
614 2021. Training verifiers to solve math word
615 problems. *arXiv preprint arXiv:2110.14168*.
- 616 Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming
617 Ma, Zhifang Sui, and Furu Wei. 2022. Why can gpt
618 learn in-context? language models implicitly perform
619 gradient descent as meta-optimizers. *arXiv preprint
620 arXiv:2212.10559*.
- 621 Atticus Geiger, Hanson Lu, Thomas Icard, and Christo-
622 pher Potts. 2021. Causal abstractions of neural net-
623 works. *Advances in Neural Information Processing
624 Systems*, 34:9574–9586.
- 625 Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot,
626 Dan Roth, and Jonathan Berant. 2021. Did Aristotle
627 Use a Laptop? A Question Answering Bench-
628 mark with Implicit Reasoning Strategies. *Transac-
629 tions of the Association for Computational Linguis-
630 tics (TACL)*.
- 631 Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-
632 Peng Lim, Lidong Bing, Xing Xu, Soujanya Po-
633 ria, and Roy Ka-Wei Lee. 2023. Llm-adapters:
634 An adapter family for parameter-efficient fine-
635 tuning of large language models. *arXiv preprint
636 arXiv:2304.01933*.
- 637 Jing Huang, Zhengxuan Wu, Christopher Potts, Mor
638 Geva, and Atticus Geiger. 2024. Ravel: Evalu-
639 uating interpretability methods on disentangling
640 language model representations. *arXiv preprint
641 arXiv:2402.17700*.
- 642 Shima Imani, Liang Du, and Harsh Shrivastava. 2023.
643 Mathprompter: Mathematical reasoning using large
644 language models. *arXiv preprint arXiv:2303.05398*.
- 645 Peter A Jansen, Elizabeth Wainwright, Steven Mar-
646 morstein, and Clayton T Morrison. 2018. Worldtree:
647 A corpus of explanation graphs for elementary sci-
648 ence questions supporting multi-hop inference. *arXiv
649 preprint arXiv:1802.03052*.
- 650 Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate
651 Kushman, and Hannaneh Hajishirzi. 2016. **MAWPS:**
652 **A math word problem repository**. In *Proceedings of
653 the 2016 Conference of the North American Chapter
654 of the Association for Computational Linguistics: Hu-
655 man Language Technologies*, pages 1152–1157, San
656 Diego, California. Association for Computational
657 Linguistics.
- 658 Jiachun Li, Pengfei Cao, Chenhao Wang, Zhuoran Jin,
659 Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao.
660 2024. Focus on your question! interpreting and miti-
661 gating toxic cot problems in commonsense reasoning.
662 *arXiv preprint arXiv:2402.18344*.
- 663 Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang,
664 Jason Eisner, Tatsunori Hashimoto, Luke Zettle-
665 moyer, and Mike Lewis. 2022. Contrastive decoding:
666 Open-ended text generation as optimization. *arXiv
667 preprint arXiv:2210.15097*.
- 668 Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri
669 Edwards, Bowen Baker, Teddy Lee, Jan Leike,
670 John Schulman, Ilya Sutskever, and Karl Cobbe.
671 2023. Let’s verify step by step. *arXiv preprint
672 arXiv:2305.20050*.
- 673 Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blun-
674 som. 2017. Program induction by rationale genera-
675 tion: Learning to solve and explain algebraic word
676 problems. *arXiv preprint arXiv:1705.04146*.
- 677 Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu,
678 Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark,
679 and Ashwin Kalyan. 2022. Dynamic prompt learning
680 via policy gradient for semi-structured mathematical
681 reasoning. *arXiv preprint arXiv:2209.14610*.
- 682 Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish
683 Sabharwal. 2018. Can a suit of armor conduct elec-
684 tricity? a new dataset for open book question answer-
685 ing. *arXiv preprint arXiv:1809.02789*.
- 686 Arkil Patel, Satwik Bhattacharya, and Navin Goyal.
687 2021. Are nlp models really able to solve
688 simple math word problems? *arXiv preprint
689 arXiv:2103.07191*.
- 690 Nazneen Fatema Rajani, Bryan McCann, Caiming
691 Xiong, and Richard Socher. 2019. **Explain your-
692 self! leveraging language models for commonsense
693 reasoning**. In *Proceedings of the 2019 Conference
694 of the Association for Computational Linguistics
(ACL2019)*.

696	Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. <i>Communications of the ACM</i> , 64(9):99–106.	751
697		752
698		753
699		754
700		755
701	Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialqa: Commonsense reasoning about social interactions. <i>arXiv preprint arXiv:1904.09728</i> .	756
702		757
703		758
704	Karen Simonyan. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. <i>arXiv preprint arXiv:1312.6034</i> .	759
705		760
706		761
707	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. <i>arXiv preprint arXiv:1811.00937</i> .	762
708		763
709		764
710		765
711	Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. <i>arXiv preprint arXiv:1806.02847</i> .	766
712		767
713		768
714	A Vaswani. 2017. Attention is all you need. <i>Advances in Neural Information Processing Systems</i> .	769
715		770
716	Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. <i>arXiv preprint arXiv:2305.14160</i> .	771
717		772
718		773
719		774
720		775
721	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. <i>Advances in neural information processing systems</i> , 35:24824–24837.	776
722		777
723		778
724		779
725		780
726	Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024a. Advancing parameter efficiency in fine-tuning via representation editing. <i>arXiv preprint arXiv:2402.15179</i> .	781
727		782
728		783
729		784
730		785
731		786
732	Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024b. Reft: Representation finetuning for language models. <i>arXiv preprint arXiv:2404.03592</i> .	787
733		788
734		789
735		790
736		791
737	Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. 2024c. <code>pyvne</code> : A library for understanding and improving PyTorch models via interventions. In <i>Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)</i> , pages 158–165, Mexico City, Mexico. Association for Computational Linguistics.	792
738		793
739		794
740		795
741		796
742		797
743		798
744		799
745		800
746		801
747	Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. <i>arXiv preprint arXiv:2309.17453</i> .	802
748		803
749		804
750		805

772 A Implement Details

773 A.1 Datasets

774 The test datasets that we use across two scenarios covering eight datasets: GSM8K (Cobbe et al.,
775 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-
776 Kedziorski et al., 2016), SVAMP (Patel et al.,
777 2021), BoolQ (Clark et al., 2019), SocialIQA (Sap
778 et al., 2019), WinoGrande (Sakaguchi et al., 2021),
779 and OpenBookQA (Mihaylov et al., 2018).

780 **GSM8K.** GSM8K, which comprises grade-school
781 math word problems requiring multi-step rea-
782 soning, usually takes between 2 and 8 steps to
783 solve problems using basic arithmetic operations
784 $+, -, \times, \div$. We used the last 300 samples in the
785 training set as the validation set and reported the
786 results on its test set.

787 **Arithmetic Reasoning Scenarios.** Following
788 the experimental setup established in Hu et al.
789 (2023), we fine-tune a combined dataset of seven
790 arithmetic reasoning tasks, called Math10K,
791 utilizing LM-generated chain-of-thought steps.
792 We report performance metrics in three test sets:
793 AQuA, MAWPS, and SVAMP.

794 **Commonsense Reasoning Scenarios.** For
795 commonsense reasoning scenarios, we opted not
796 to use Commonsense170K from Hu et al. (2023),
797 as it does not incorporate CoT steps. We create a
798 suitable Commonsense60k training set, combining
799 six commonsense reasoning tasks: CommonsenseQA
800 (Talmor et al., 2018), CoS-e (Rajani et al.,
801 2019), OpenBookQA (Mihaylov et al., 2018),
802 SocialIQA (Sap et al., 2019), StrategyQA (Geva
803 et al., 2021), WorldTree (Jansen et al., 2018). We
804 report performance metrics in four test sets: BoolQ,
805 SocialIQA, WinoGrande, and OpenBookQA.

807 A.2 Base Models

808 We finetune our models on LLaMA-2-7B, LLaMA-
809 2-13B, LLaMA-3-8B and Mistral-7B. We use the
810 “chat” version of LLaMA-2, and the “instruct” ver-
811 sion of LLaMA-3 and Mistral-7B.

812 A.3 Hyperparameters

813 For a fair comparison with ReFT ($p7 + s7$), we
814 selected 14 intervention representations and main-
815 tained a rank of 8, consistent with the parameters
816 used in ReFT. We set the hyperparameters of α to
817 0.05. And we used the “order” selection criteria by
818 default. To ensure a fair comparison, we maintain
819 the same training principle, details are shown in

Table 9: The values of hyperparameters.

HyperParameters	Values
Rank	8
Number of representations	14
threshold α	0.05
selection criteria	Order(default)
Number of Epochs	12 for arithmetic, 6 for commonsense
Batch Size	2
Gradient accumulation steps	16
seed	42
Optimizer	AdamW
Learning Rate Schedule	Linear
Learning Rate	$9e - 4$
dropout	0.05 for GSM8K, 0 for others
Weight Decay	0.06 for GSM8K, 0 for others
Warmup ratio	0 for GSM8K, 0.1 for others

820 Table 9. For all tasks, model outputs are generated
821 with greedy search.

822 A.4 Prompt

823 We use a prompt for each task.

824 **GSM8K**
[question]
Answer the above question. First, think step
by step and then answer the final number.

825 **Other Arithmetic Scenario**
Below are instructions for a task. Write a
response that appropriately completes the
request.
Instruction:
[Question]
Response:

826 **Commonsense Scenario**
[Question]
the correct answer is

827 B More models

828 We verify our method, CRFT, in different mod-
829 els. We tested four basic models (LLaMA-2-7B,
830 LLaMA-2-13B, LLaMA-3-8B, Mistral-7B) on the
831 GSM8K dataset, covering different sizes and fam-
832 ilies. The consistent improvement of our exper-
833 imental results demonstrates the effectiveness of
834 our method.

Table 10: **Quantitative comparison on GSM8K with four base models: LLaMA-2-7B, LLaMA-2-13B, LLaMA-3-8B, and Mistral-7B.** The best performance is highlighted in **bold**, while the second-best is underlined.

PEFT	Identify	Accuracy (\uparrow)			
		LLaMA-2-7B	LLaMA-2-13B	LLaMA-3-8B	Mistral-7B
None	-	14.6	30.9	64.5	38.4
ReFT	p7+s7	29.0	37.9	64.7	46.5
	SAF	30.4 29.6	38.7 39.6	<u>70.8</u> 70.6	46.4 46.9
	MAF	32.0 <u>32.1</u>	38.3 38.0	67.5 64.8	48.0 47.3
CRFT (ours)	Union(attn)	31.2 32.8	40.3 39.4	64.4 71.0	<u>48.1</u> 47.7
	SSF	31.4 30.4	<u>40.1</u> 38.4	64.6 64.5	46.4 46.5
	MSF	31.4 30.3	38.3 38.3	64.5 65.1	46.9 47.7
	Union(sal)	32.8 31.5	38.8 38.3	63.8 64.0	48.0 48.2

835 C Generation examples

836 We give a generation example for each test dataset.

837 D Attention Analysis

838 We visualize the attention score of all 32 heads
 839 in the final layer and the last head in all layers,
 840 which illustrates that our method indeed enriches
 841 information interactions.

Instruction (GSM8K)

Marie ordered one chicken meal that costs \$12, 5 packs of milk that costs \$3 each, 4 apples that cost \$1.50 each, and some boxes of pizza. Marie paid a total of \$50. How many boxes of pizza did Marie order if each box costs \$8.50?

Response

Model	Response
LLaMA-2-7B	<p>Great, let's break it down step by step to find out how many boxes of pizza Marie ordered:</p> <ol style="list-style-type: none"> 1. First, let's calculate the total cost of Marie's meal: \$12 (chicken meal) + \$15 (5 packs of milk) = \$27 2. Next, let's calculate the total cost of Marie's fruits: \$1.50 (4 apples) = \$6 3. Now, let's add the total cost of Marie's meal and fruits: \$27 + \$6 = \$33 4. Finally, let's subtract the total cost of Marie's meal and fruits from the total amount she paid: \$50 - \$33 = \$17 <p>So, Marie ordered \$17 worth of pizza boxes, which is equal to 4 boxes (since each box costs \$8.50). Therefore, the answer is 4 boxes of pizza.</p>
ReFT	<p>Marie paid a total of $\\$12 + \\$3 \times 5 = \\$25$ for the milk. She paid a total of $\\$15 + \\$1.50 \times 4 = \\$19$ for the apples. So, Marie paid a total of $\\$50 - \\$25 - \\$19 = \\16 for the pizza. Therefore, Marie ordered $\\$16 / \\$8.50 = 1.90$ boxes of pizza. The final answer is: 1.9</p>
CRFT	<p>The cost of 5 packs of milk is $\\$3 \times 5 = \\15. The cost of 4 apples is $\\$1.50 \times 4 = \\6. The total cost of the chicken meal, milk, and apples is $\\$12 + \\$15 + \\$6 = \\33. So, Marie ordered $\\$50 - \\$33 = \\$17$ worth of pizza. Therefore, Marie ordered $\\$17 / \\$8.50 = 2$ boxes of pizza. The final answer is: 2</p>

Instruction (AQuA)

Two ants are standing side-by-side. One ant, which is 4 inches tall, casts a shadow that is 10 inches long. The other ant is 6 inches tall. Compute, in inches, the length of the shadow that the taller ant casts. Answer Choices: (A) 36 (B) 28 (C) 42 (D) 15 (E) 20

Response

Model	Response
LLaMA-2-7B	<p>Great, I'd be happy to help you with that!</p> <p>According to the instruction, the length of the ribbon was originally 30 cm and it was reduced in the ratio 5 : 3. To find the length of the ribbon now, we can use the ratio to find the new length.</p> <p>The ratio of the original length to the new length is 5 : 3, which means that for every 5 cm of the original length, there are 3 cm of the new length.</p> <p>So, if the original length of the ribbon was 30 cm, the new length would be:</p> $30 \text{ cm} \times \frac{3}{5} = 20 \text{ cm}$ <p>Therefore, the length of the ribbon now is 20 cm.</p> <p>The correct answer is (B) 30 cm.</p>
ReFT	<p>Let's call the length of the shadow cast by the taller ant "x".</p> <p>We know that the shadow of the shorter ant is 10 inches long, so we can set up the equation:</p> $x + 4 = 10$ <p>To solve for "x", we can subtract 4 from both sides:</p> $x = 10 - 4$ $x = 6$ <p>Therefore, the length of the shadow cast by the taller ant is 6 inches.</p> <p>The answer is (C) 42.</p>
CRFT	<p>To solve this problem, we need to use the formula for the length of a shadow:</p> $\text{Shadow Length} = (\text{Shadow Distance}) \times (\text{Shadow Ratio})$ <p>where Shadow Distance is the distance between the light source and the object casting the shadow, and Shadow Ratio is the ratio of the shadow distance to the height of the object casting the shadow.</p> <p>In this problem, the light source is the sun, and the distance between the light source and the first ant is 10 inches. The height of the first ant is 4 inches, so the Shadow Ratio is:</p> $\text{Shadow Ratio} = \text{Shadow Distance} / \text{Height of Object}$ $\text{Shadow Ratio} = 10 / 4$ $\text{Shadow Ratio} = 2.5$ <p>To find the length of the shadow that the taller ant casts, we need to multiply the Shadow Ratio by the height of the taller ant:</p> $\text{Shadow Length} = 2.5 \times 6$ $\text{Shadow Length} = 15 \text{ inches}$ <p>Therefore, the answer is (D) 15.</p>

Instruction (MAWPS)

The distance measured on a map varies directly with the actual distance . If 1 cm represents 25 kilometers , how many kilometers are represented by 7 cm ?

Response

Model

LLaMA-2-7B

Response

Great, I'm happy to help! Based on the instruction you provided, if 1 cm on a map represents 25 kilometers, then we can conclude that the ratio of the distance represented on the map to the actual distance is 1:25.
So, if we want to find out how many kilometers are represented by 7 cm on the map, we can simply multiply the ratio by the number of cm given:
 $7 \text{ cm} \times 1 \text{ km}/25 \text{ km} = 0.28 \text{ km}$
Therefore, 7 cm on the map represents approximately 0.28 kilometers.

ReFT

We can set up a proportion to solve this problem:
 $1 \text{ cm} / 25 \text{ km} = 7 \text{ cm} / x \text{ km}$
To solve for x, we can cross-multiply:
 $1 \text{ cm} * 25 \text{ km} = 7 \text{ cm} * x \text{ km}$
Multiplying the two sides gives us:
 $25000 \text{ cm} = 7x \text{ km}$
Dividing both sides by 7 gives us:
 $x \text{ km} = 3500 \text{ cm}$
Therefore, 7 cm represents 3500 kilometers.
Answer: 3500

CRFT

We can use the concept of direct variation to solve this problem.
Let x be the actual distance represented by 7 cm on the map.
We know that:
1 cm on the map represents 25 km
So, 7 cm on the map represents:
 $7 \times 1 = 7 \times 25 = 175 \text{ km}$
Therefore, 7 cm on the map represents 175 kilometers.
Answer: 175

Instruction (SVAMP)

A mailman has to give 38 pieces of junk mail to each of the 78 blocks. If there are 19 houses on a block. How many pieces of junk mail should he give each house?

Response

Model

LLaMA-2-7B

Response

Great, I'm happy to help! Based on the instruction you provided, here's the calculation:
To determine how many pieces of junk mail the mailman should give each house, we need to divide the total number of pieces of junk mail (38) by the number of houses on each block (19).
So, the mailman should give:
 $38 \text{ pieces of junk mail} / 19 \text{ houses per block} = 2 \text{ pieces of junk mail per house}$
Therefore, the mailman should give 2 pieces of junk mail to each of the 78 blocks, for a total of 156 pieces of junk mail.

ReFT

To solve this problem, we need to divide the total number of junk mail by the number of houses per block:
 $38 \text{ junk mail pieces per block} \times 19 \text{ houses per block} = 724 \text{ junk mail pieces}$
Therefore, the mailman should give 724 junk mail pieces to each house.
Answer: 724

CRFT

To solve this problem, we need to divide the total number of junk mail by the number of houses per block:
 $38 / 19 = 2$
Therefore, the mailman should give 2 pieces of junk mail to each house.
The answer in Arabic numerals is: 2.

Instruction (BoolQ)

Please answer the following question with true or false, question: have scotland ever been in the world cup final?
Answer format: true/false

Response

Model	Response
LLaMA-2-7B	True. Northern Ireland is part of the United Kingdom, which includes England, Scotland, and Wales, as well as Northern Ireland. The term "Great"
ReFT	northern ireland is part of great britain. Therefore, the answer is: true/false true
CRFT	scotland has never been in the world cup final Therefore, the answer is false.

Instruction (SQuAD)

Please choose the correct answer to the question: Riley had twisted their leg and began to hurt so bad that they could not walk. What does Riley need to do before this?
Answer1: lie down Answer2: run Answer3: Sit and relax
Answer format: answer1/answer2/answer3

Response

Model	Response
LLaMA-2-7B	Based on the information provided in the question, the correct answer is: Answer3: Sit and relax. Before Riley can't
ReFT	the answer is: answer1
CRFT	The answer is: answer2: run

Instruction (WinoGrande)

Please choose the correct answer to fill in the blank to complete the given sentence: I had to read an entire story for class tomorrow. Luckily, the _ was canceled.
Option1: story Option2: class Answer format: option1/option2

Response

Model	Response
LLaMA-2-7B	Great, I'm happy to help! The correct answer is: Option1: story So, you had to read an entire story
ReFT	the toaster oven is too small to hold the eggplant. Therefore, the answer is: Option2: toaster
CRFT	class is the only option that can be canceled. Therefore, the answer is: class

Instruction (OBQA)

Please choose the correct answer to the question: An example of conservation is avoiding the use of
Answer1: gasoline Answer2: air Answer3: snow Answer4: clothes
Answer format: answer1/answer2/answer3/answer4

Response

Model	Response
LLaMA-2-7B	Great, I'm happy to help! An example of conservation is avoiding the use of: Answer4: clothes. Conserv
ReFT	Great, I'm happy to help! An example of conservation is avoiding the use of: Answer4: clothes Conservation
CRFT	The answer is: answer1

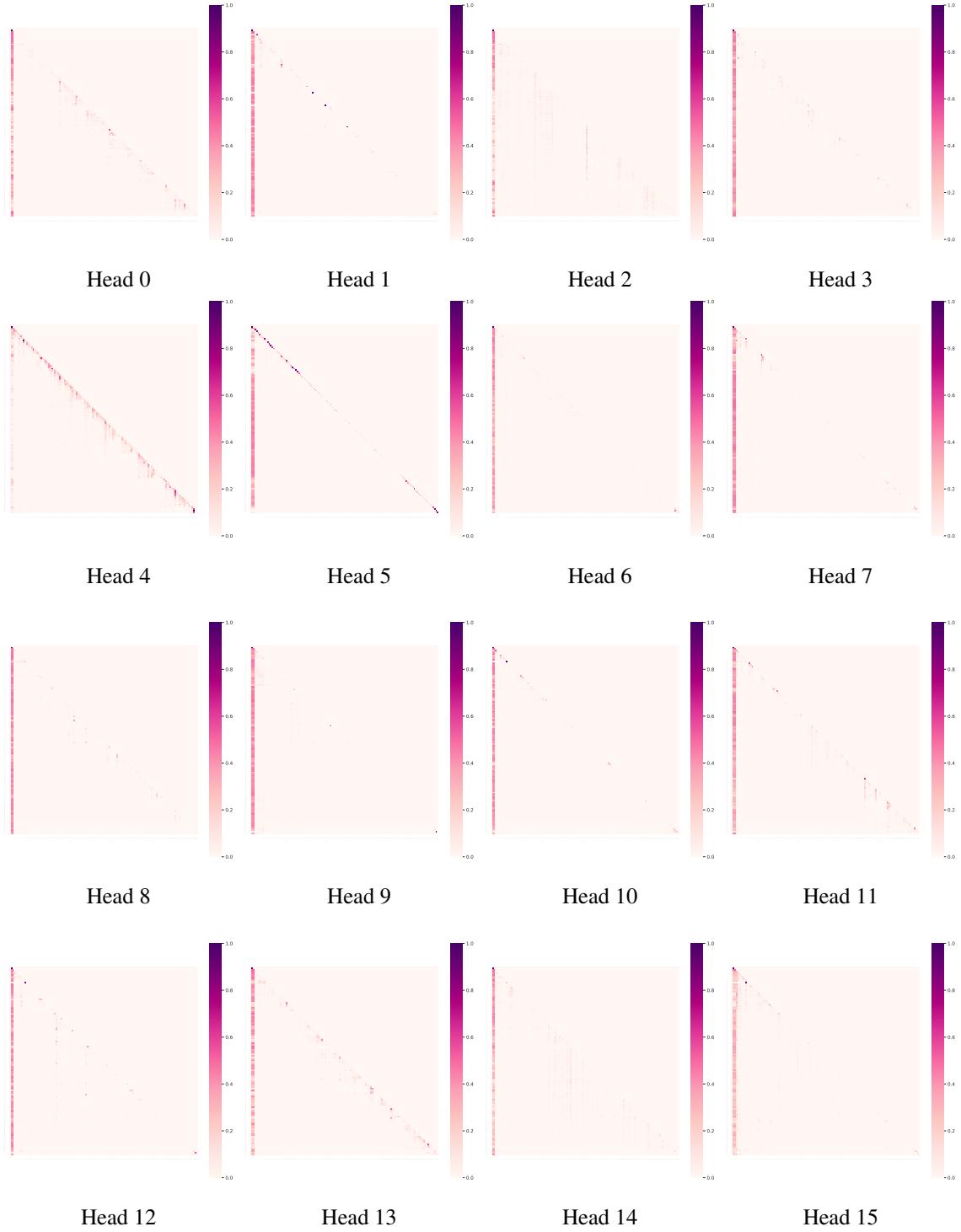


Figure 6: **The attention score of LLaMA-2-7B in layer 31.** (part 1 of 2)

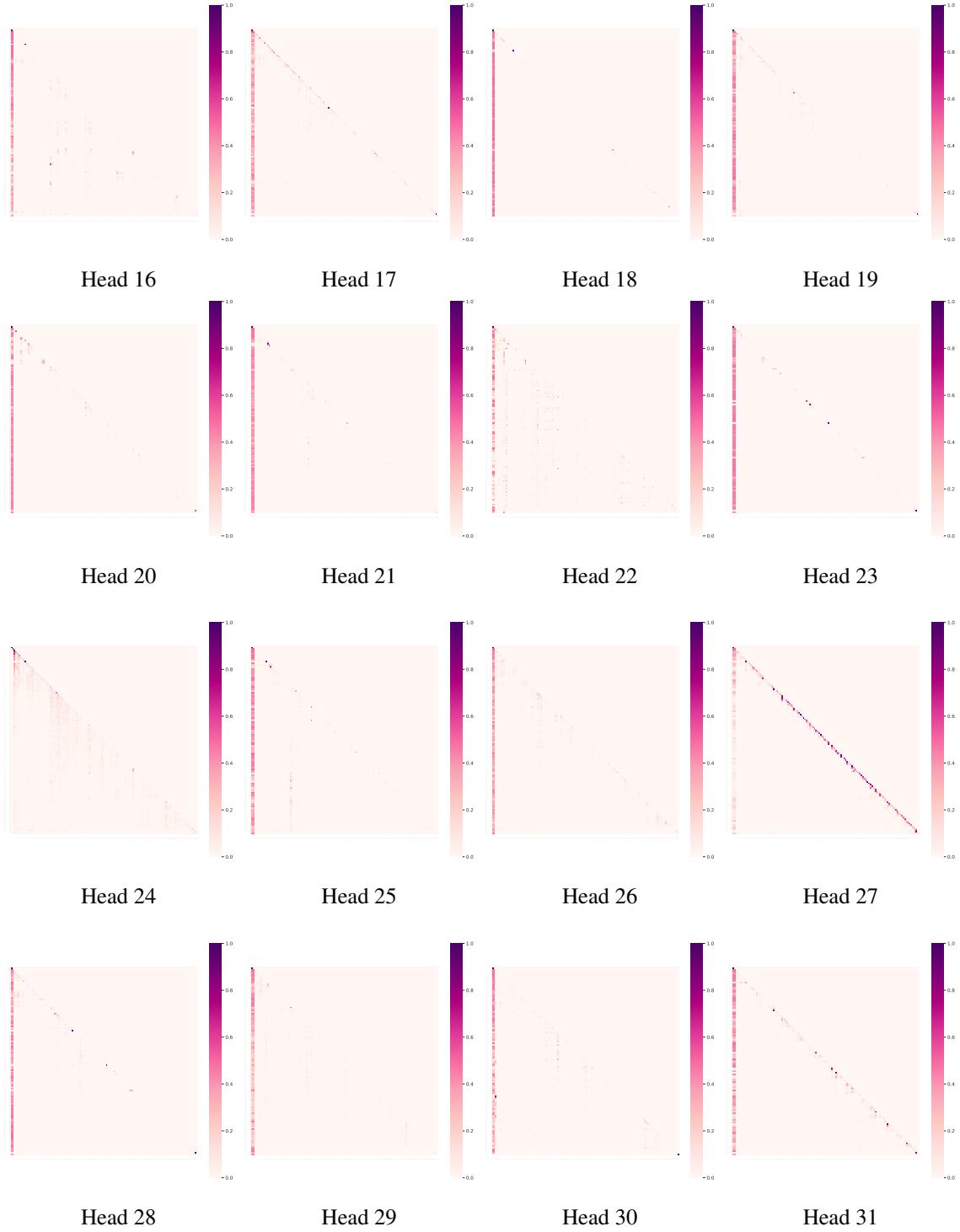


Figure 7: The attention score of LLaMA-2-7B in layer 31. (part 2 of 2)

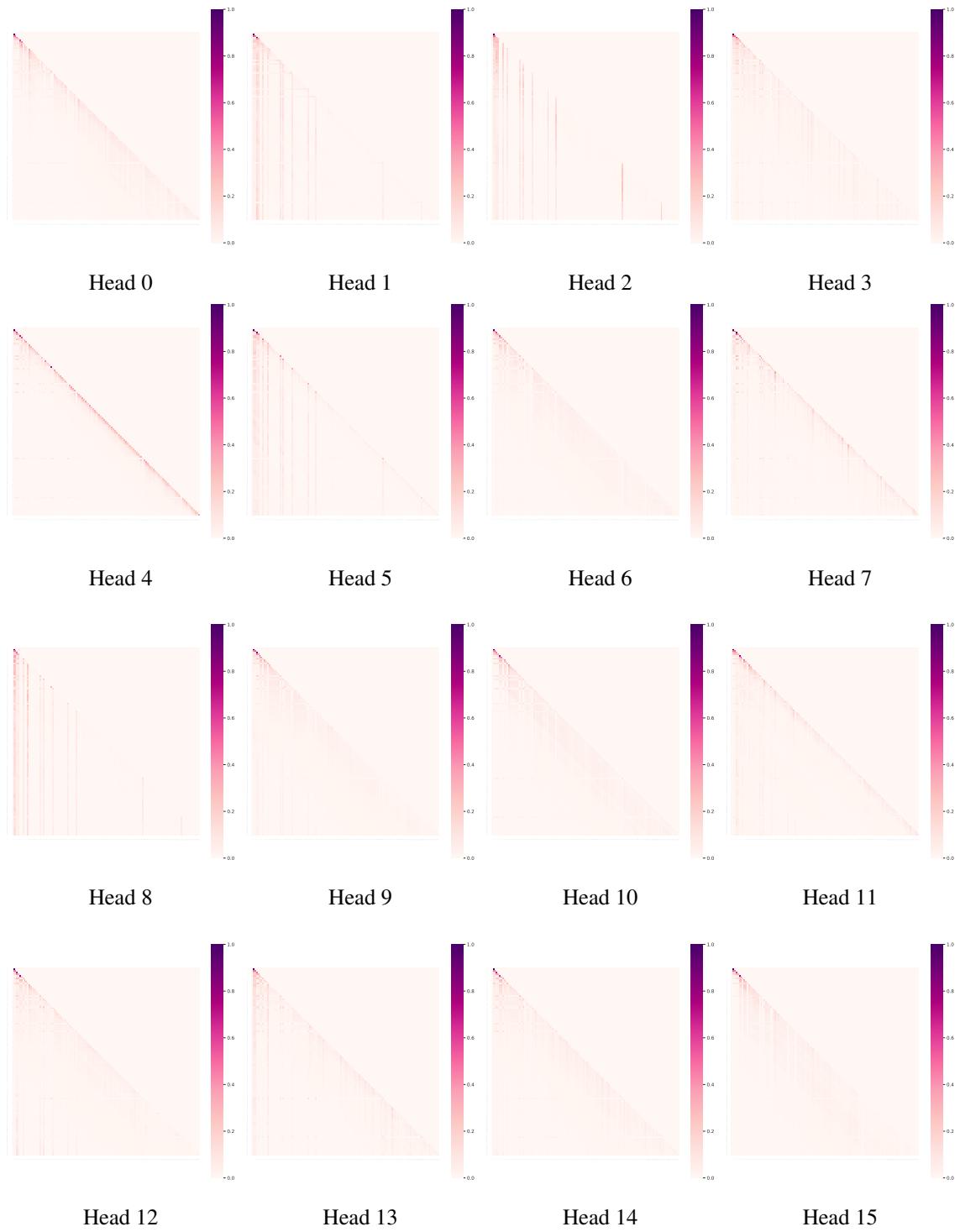


Figure 8: **The attention score of our CRFT in layer 31.** (part 1 of 2)



Figure 9: **The attention score of our CRFT in layer 31.** (part 2 of 2)

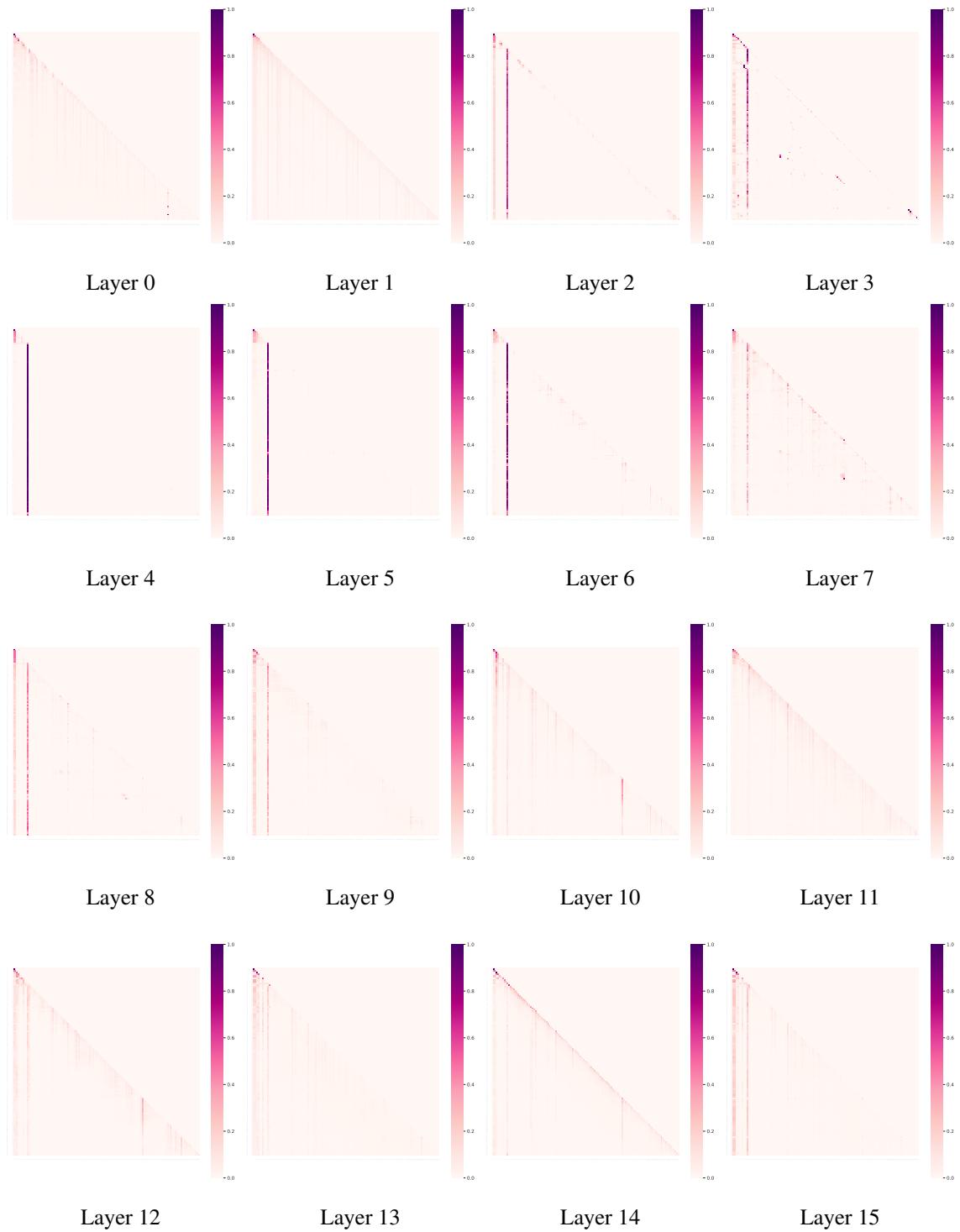


Figure 10: **The attention score of our CRFT on head 31 in all layers.** (part 1 of 2)

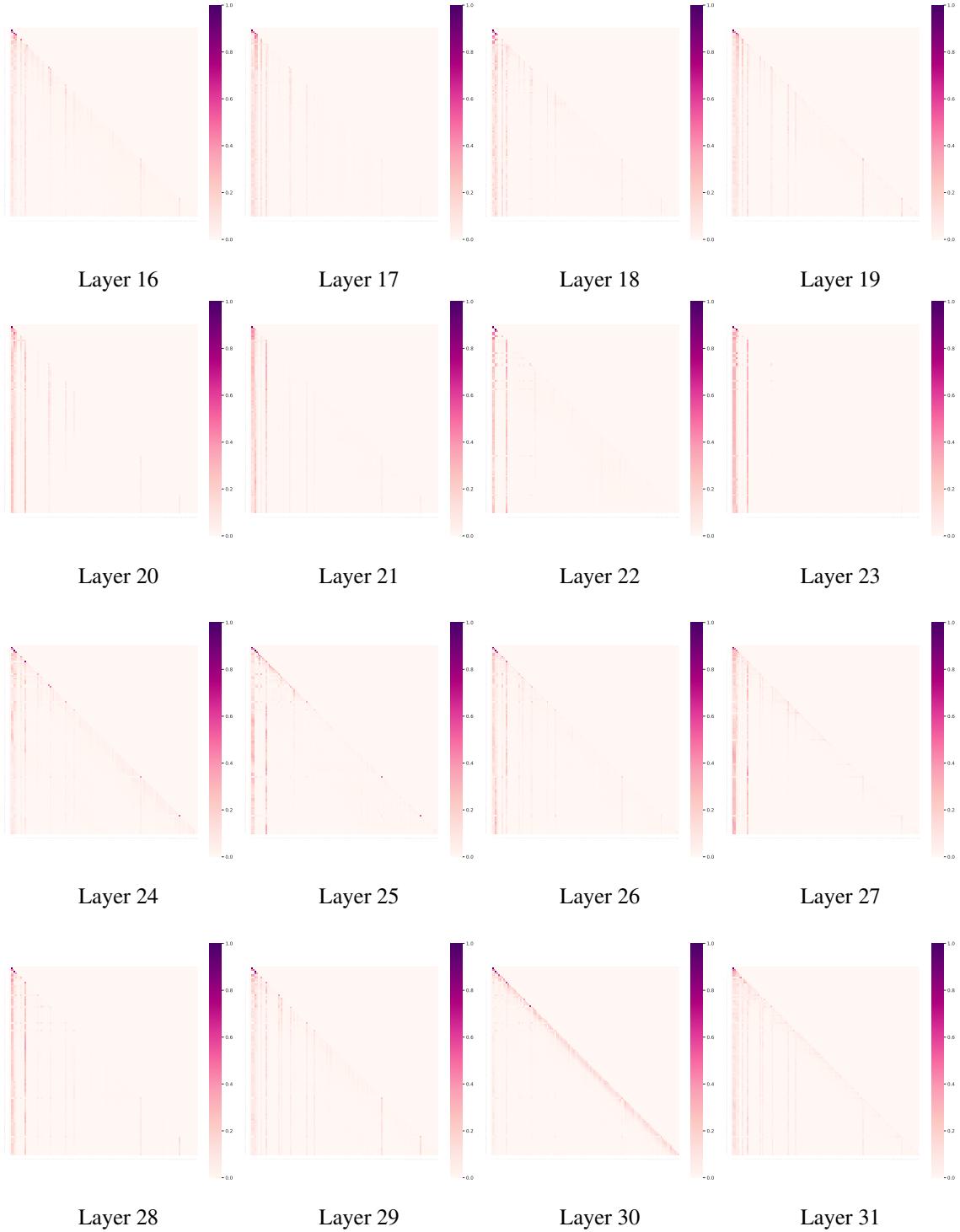


Figure 11: **The attention score of our CRFT on head 31 in all layers.** (part 2 of 2)